

УДК 371.3

О. І. Гороховський, к. т. н., доц.; Т. І. Трояновська

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПОБУДОВИ АДАПТИВНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ

Стаття присвячена опису розробки інформаційної технології побудови адаптивних систем дистанційного навчання з урахуванням статичної та динамічної складових предметно-орієнтованої домінанти студента, а також з урахуванням індивідуального часового графіку та швидкості сприйняття навчального матеріалу студентом.

Ключові слова: інформаційна технологія, адаптивна система дистанційного навчання, предметно-орієнтована домінанта.

Вступ

Для надання студенту дистанційної форми навчання (ДФН) можливості отримувати якісну дистанційну освіту Г. Вебер [1] пропонує додати до системи дистанційного навчання (СДН) певну адаптивність. Адаптивна СДН (АСДН) “підлаштовується” до особливостей навчання окремого студента. Такий напрямок розробок наразі є **актуальним**, оскільки складність дисциплін (особливо спеціалізованих) зростає і скласти певний усереднений план подачі матеріалу для них надзвичайно складно. Це, у свою чергу, знижує загальний рівень якості освіти і складає вагомую частину в переліку проблем сучасної освітньої системи.

Метою цієї статті є розробка інформаційної технології побудови АСДН, здатної пристосовуватися до особливостей сприйняття навчального матеріалу студентом. Для досягнення поставленої мети система повинна містити компоненти збору, обробки, зберігання даних та їх використання під час видачі студенту матеріалів дистанційного курсу (ДК). Для цього до АСДН включено такі модулі: модуль збору даних про діяльність студента (для забезпечення первинних даних за прямими і непрямыми оцінками [2]); модуль видачі методичних матеріалів для безпосереднього відображення лекційних і практичних матеріалів; автоматичний модуль обробки даних і видачі матеріалу (АМОД) для обробки первинних даних оцінювання і відповідного підбору навчальних матеріалів і контрольних елементів.

Технологія розробки системи, яка включатиме такі модулі, має наступні етапи: розробка структур даних; розробка АРМ викладача; розробка АМОД (підсистеми видачі студенту матеріалів і підсистеми аналізу даних та коригування курсу); розробка АРМ студента. Діаграма відповідного технологічного процесу показана на рисунку 1.

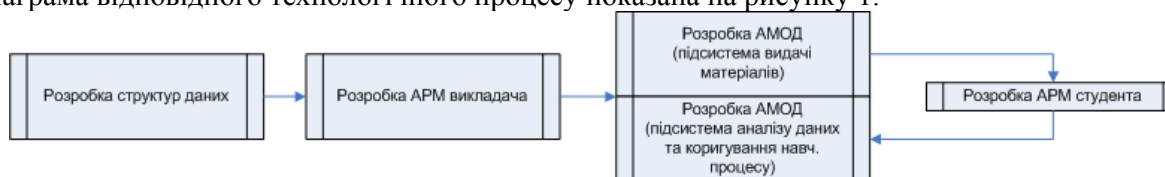


Рис. 1. Діаграма технологічного процесу розробки АСДН

Розробка АМОД та розробка АРМ студента показані на цьому рисунку як циклічний процес, оскільки вони взаємопов'язані.

1. Розробка структур даних

Загалом, СДН базується на представленні матеріалів, що входять до складу курсів, які пропонуються студентам. Аналізуючи сучасні електронні підручники (ЕП), можна сказати, що описати ДК можна за допомогою файлу формального змісту. АСДН може використати цей файл для визначення поточного місцезнаходження студента в ДК та для

відслідковування шляху користувача. Виділяють лінійний та ієрархічний формати, які визначають зміст курсу:

Лінійний формат. Використовується для реалізації ДК за традиційною схемою ЕП із лінійною компоновкою матеріалу (рисунок 2).

Ієрархічне представлення. Крім елементів курсу, містить вкладені категорії (модулі, теми, лекції), на які логічно поділений ДК (рисунок 3).



Рис. 2. Схема лінійного представлення

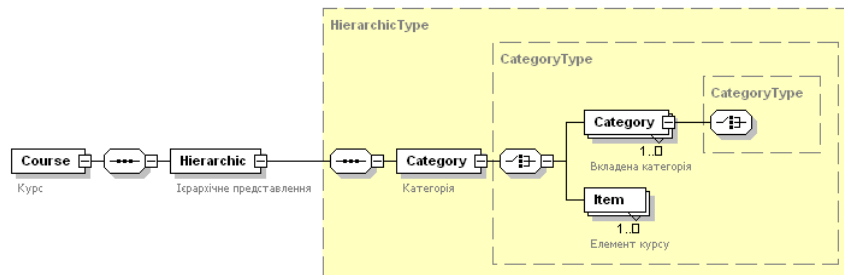


Рис. 3. Схема ієрархічного представлення курсу

Для розгортання ДК достатньо існування файлів одного з вищеназваних типів і гарантованого існування кожного елементу курсу. Для забезпечення цілісної передачі курсу, розробленого в автономному режимі до АСДН, розроблено модуль генерації ЕП для переведення матеріалу в необхідний формат – АРМ викладача.

2. АРМ викладача

Розробка ДК полягає у створенні структурних одиниць, з яких організовано курс. Оскільки навчальний матеріал подається студенту дозовано відповідно його рівню підготовки та перебігу його навчання, введемо додаткове визначення «поняття» – логічно завершеного фрагменту навчального матеріалу.

Технологічний процес розробки ДК полягає у виконанні наступних дій: 1) розробки плану курсу; 2) інформаційного наповнення плану курсу; 3) розбиття матеріалу на окремі поняття; 4) формування послідовності структурних елементів курсу з понять відповідно до створеного плану курсу.

Для забезпечення цього технологічного процесу необхідно виокремити основні структурні елементи курсу. Вони можуть відповідати за потребою як традиційній (статичній), так і динамічній компоновці ЕП: модуль, тема, лекція, поняття. У цій розробці використовується динамічне компоновання навчального матеріалу, оскільки для того, щоб забезпечити студенту подачу матеріалу відповідно його рівня, ця компоновка повинна формуватися та коригуватися в процесі навчання.

Отже, узагальнена технологічна процедура виглядатиме так:

- 1) Розбиття масиву інформації з дисципліни на окремі поняття.
- 2) Логічне групування понять.
- 3) Інформаційне наповнення поняття. Для кожного поняття мають бути передбачені різні варіанти подачі матеріалу.
- 4) Групування понять у лекції відповідно логічному групуванню.
- 5) Побудова логічної послідовності лекцій (проекту курсу).
- 6) Групування лекцій в теми та модулі.

- 7) Створення єдиного індекс-файлу, який містить зміст курсу у формі, що може бути видана на АРМ студента. Одразу зауважимо, що цей файл може змінюватись відповідно до перебігу навчання студента, на відміну від зміст-файлу, який створюється на етапі проектування ДК.
- 8) Створення пакету курсу (наприклад, у вигляді Zip-файлу за стандартом SCORM).

Для реалізації такого технологічного ланцюжка призначено АРМ викладача, на вхід якого подається набір матеріалів, а на виході маємо логічно організований ДК, який можна розгортати на сервері. Серверна частина АМОД приймає створений курс і подає його студенту.

3. Розробка АМОД: підсистема видачі матеріалів

АМОД розгортається на сервері і є набором спеціалізованих програм, які обробляють запити користувача, або інших програм, що звертаються до нього. Загальна функціональна модель роботи АМОД показана на рисунку 4.

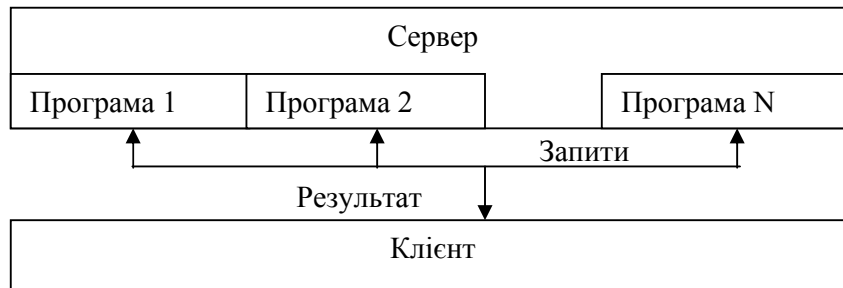


Рис. 4. Загальна функціональна модель роботи АМОД

Для реалізації серверних програм обробки даних була обрана технологія сервлетів [3]. Агент збору даних приймає і надсилає запити в XML-форматі. HTML-запити можуть бути двох варіантів – get-запит і post-запит (рисунок 5). Get-запит використовується для перенаправлення студента на службові сторінки профілю або для забезпечення роботи агента контролю сесії. Post-запит посилається у вигляді набору значень «ключ-значення», це зручно для надсилання результатів тестування, наприклад, де кожному питанню відповідає умовне ім'я, а значенням є відповідь студента [3, 4]. Сервлет має абсолютний доступ до ресурсів сервера.

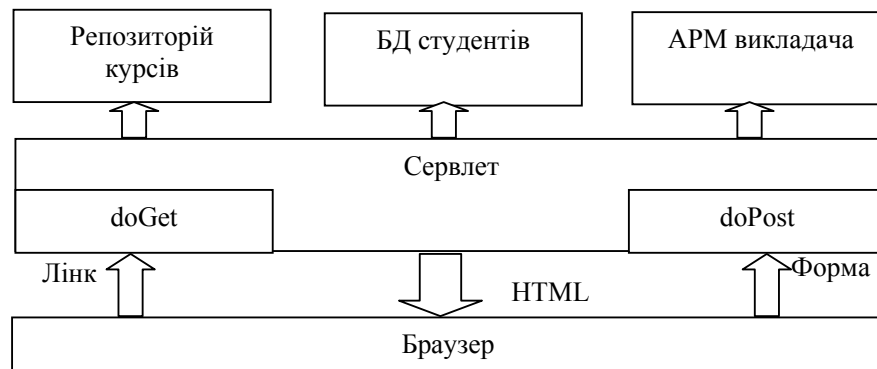


Рис. 5. Структурна модель АМОД, реалізованого на сервлетах

Сервлети відповідають за реалізацію предметно-орієнтованої домінанти (ПОД) студента [5], реалізуючи його динамічну частину. Первинні дані, зібрані модулем збору даних, через Post-запит надходять до сервлету та БД студентів.

4. Розробка АРМ студента

Завданням АРМ студента є збір даних про діяльність студента під час навчання, що є вхідними даними для підсистеми коригування курсу і є динамічною частиною ПОД [6]. Модуль збору даних є автономною програмою АРМ студента й передає дані про перебіг навчання студента на АМОД. На рисунку 6 наведено схему опису процесу роботи модуля збору даних. Sender – модуль активного компонента, який відповідає за пересилання сигналу сервлету. Scanner слідує за подіями на стороні користувача. Формат даних, за допомогою якого надсилаються сигнали про виникнення певної події на стороні користувача, показано на рис. 7. Далі переданий сигнал подається на селектор, що зчитує сигнал, декодує його і визначає, який саме сервлет відповідає за обробку цього сигналу за таблицею відповідності.

Опишемо сигнали, які відповідають подіям, що відбулися на стороні користувача, і призначимо їм відповідні сервлети: Bound Start Time – здійснює запис у БД стартової позначки таймера при переході на сторінку або на початку роботи з матеріалами лекції; Bound End Time – здійснює запис у БД кінцевої позначки таймера; Perform Test – обробляє дані, отримані під час проходження студентом тестового завдання, здійснює їх аналіз. Різні сигнали передбачають внесення змін в різні частини БД та в динамічну частину ПОД.

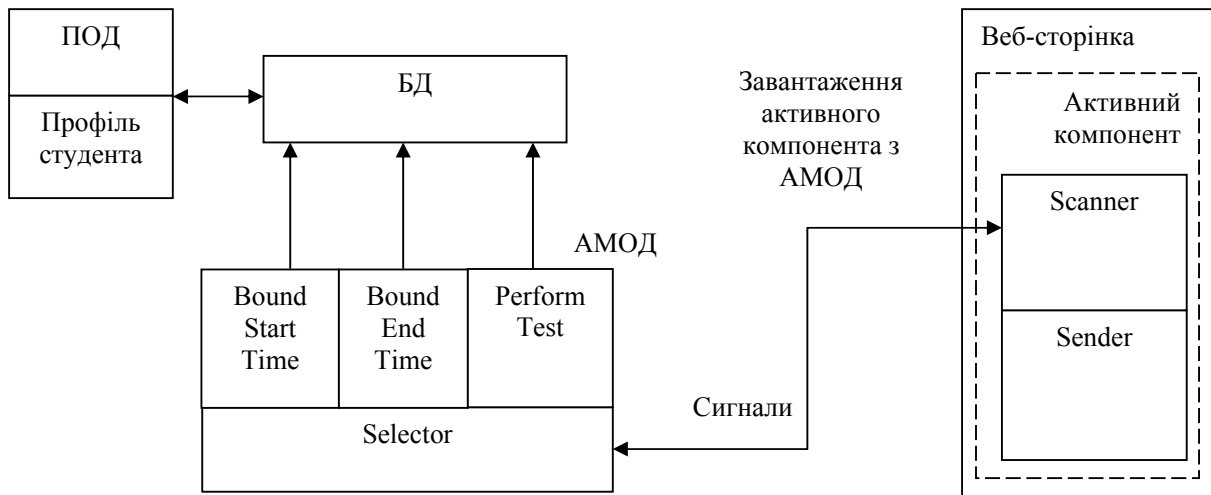


Рис 6. – Схема функціонування модуля збору даних

За своєю структурою аплет – це фонові нитка процесу всередині браузера, який виділяється окремий фрагмент сторінки, де розміщено користувацький інтерфейс аплету. У мові розмітки HTML для цього передбачено спеціалізований тег `applet`, який описує розташування такого компонента [4, 7]. Запуск аплету відбувається одночасно із завантаженням веб-сторінки на робочу станцію користувача, а зупинка процесу настає після того, як сторінка закривається, або відбувається перехід на наступну. Під час перебування користувача на сторінці компонент аплету має необмежений доступ до контенту сторінки, може реагувати на дії користувача через обробники подій та логічне ім'я засобами спеціалізованого об'єкта `JSObject`.

Наступним етапом після зняття даних агентом збору даних є пересилання повідомлень. З метою полегшення роботи із надсиланням та отриманням даних у якості носія доцільно використати формат XML. На рисунку 7 показано схему повідомлення, яке надсилається до АМОД.

Цей формат можна розширити, змінивши схему документа, тут від початку закладено можливість ефективного додавання нових можливостей без додаткових витрат.

Оскільки більшу частину своєї діяльності студент проводить в інтернет-браузері, користуючись ним як основним інструментом роботи із матеріалами ДК, очевидно, що АРМ студента має бути реалізовано за допомогою таких технологій, які б дозволяли інтегрувати все необхідне в інтернет-браузер.

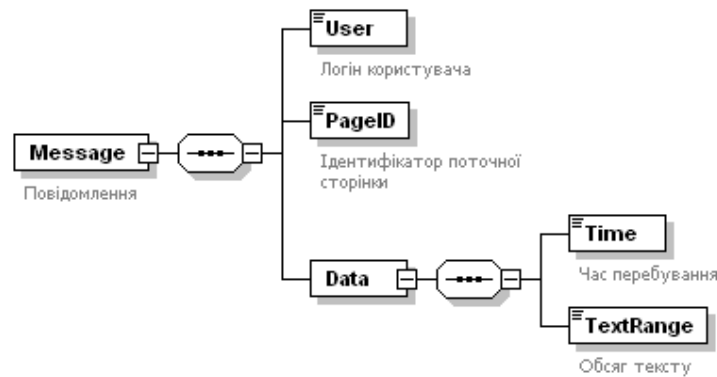


Рис. 7. Схема повідомлення, яке надсилається до АМОД

Статична частина АРМ (профіль), яка відповідає статичній частині ПОД, реалізована у вигляді веб-сторінки. Динамічна частина ПОД – частково покладена на АМОД, а частково реалізована за допомогою аплету. ПОД поділено на дві частини для забезпечення ефективності роботи з даними про перебіг навчання студента. Статичні дані, які постійні упродовж великого проміжку часу, зберігаються в окремій захищеній таблиці даних (на рис. 8 – БД користувачів), що однозначно ідентифікує студента в системі. Динамічні дані зберігаються в БД, де вони можуть бути легко відкориговані відповідно до перебігу навчання.

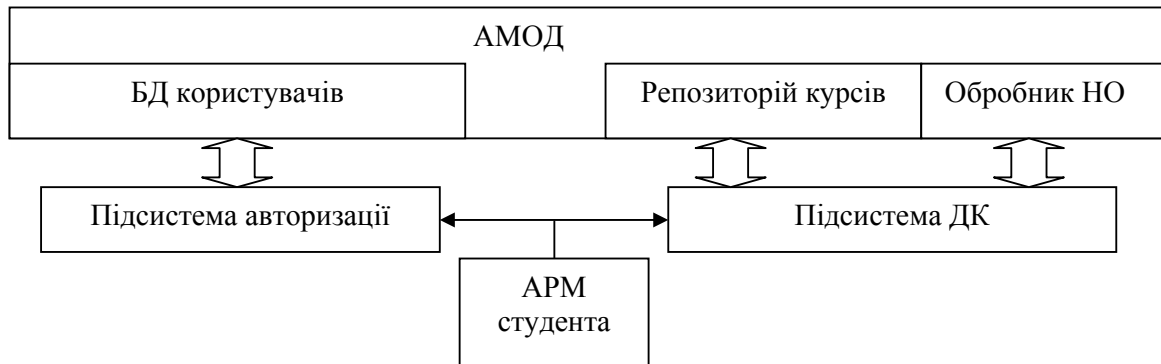


Рис. 8. Взаємодія АРМ студента із АМОД

При реєстрації студента в АСДН підсистема авторизації звертається до таблиць статичної частини ПОД і здійснює ідентифікацію студента. Під час подальшої роботи підсистема ДК, ґрунтуючись на встановленій ідентифікації, виконує задачі подачі студенту навчального матеріалу та обробки первинних даних непрямого оцінювання (НО), які є первинними даними для динамічної ПОД, яка є механізмом коригування навчального процесу.

5. Розробка АМОД: підсистеми аналізу даних та коригування курсу

Для завершення технологічного циклу розробки системи АСДН слід додати до АМОД підсистему коригування курсу. На рисунку 9 наведено загальну схему АСДН.

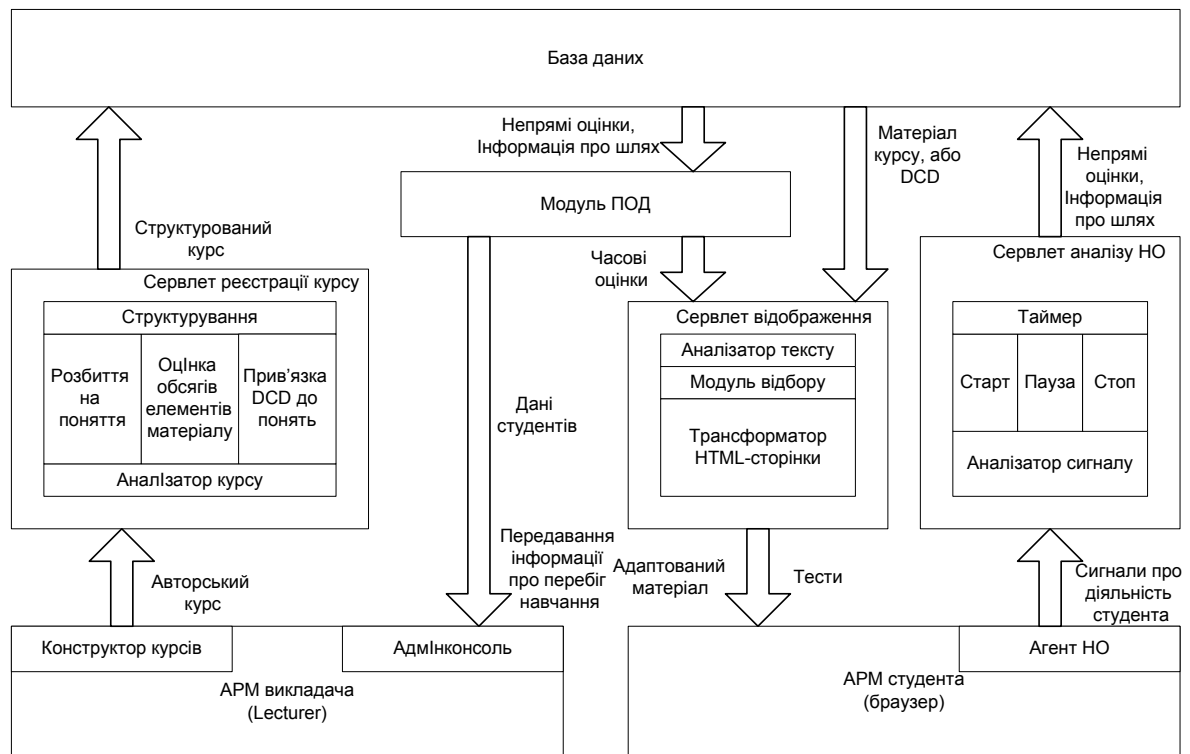


Рис. 9. Технологічна діаграма АСДН

Архітектура АМОД, заснована на сервлетах, дозволяє організувати модуль аналізу даних і коригування курсу у вигляді сервлету, на вхід якого подаються дані про перебіг навчання студента. На основі таких повідомлень формується інформація про шлях студента та НО, які вносяться до динамічної частини ПОД, а отже – впливають на спосіб подачі навчального матеріалу студентами.

Висновки

У цій статті запропоновано інформаційну технологію розробки АСДН, яка дозволяє подавати матеріал студентам з урахуванням особливостей перебігу їх навчання за допомогою використання технологічного рішення на основі сервлетів та агента збору даних, розташованих на стороні користувача. Поєднуючи наведені в статті технологічні рішення складових, отримано цілісну технологію побудови АСДН, яка дозволяє створювати адаптивне середовище для надання студентам послуг дистанційної форми навчання.

Особливості запропонованої інформаційної технології:

- 1) Динамічне формування наповнення курсу відбувається відповідно до перебігу навчання окремого студента, завдяки членуванню навчального матеріалу на окремі поняття, які потім групуються в структурні елементи курсу.
- 2) Зменшення навантаження на користувацьку сторону за рахунок використання сервлетів як основи реалізації АСДН.
- 3) Універсальність цієї системи за рахунок використання стандартів XML та HTML як основних носіїв даних.

СПИСОК ЛІТЕРАТУРИ

1. Gerhard W., Marcus S. User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems // User Modeling: Processings of the Sixth International Conference. Vienna: CISM, 1997.
2. Гороховський О.І., Трояновська Т.І., Кисюк Д.В. Автоматизація роботи викладача дистанційної форми навчання за допомогою непрямих оцінок // «Наукові дослідження – теорія та експеримент». Полтава, 14 - 16 травня, 2007. С. – 127-131.

3. Hunter J. , Crawford W. Java Servlet Programming. O'Reilly. 1998.
4. Musciano C., Kennedy B. / HTML: The Definitive Guide, 3rd Edition. O'Reilly. 1998.
5. Трояновська Т.І. Розробка комп'ютерної підсистеми аналізу та формування предметно-орієнтованої домінанти студента системи дистанційного навчання // Вісник Черкаського державного технологічного університету. - 2007. - № 3-4. - С. 41-46.
6. Гороховський О.І., Трояновська Т.І., Снігур А.В. Динамічна складова предметно-орієнтованої домінанти студента дистанційної форми навчання // Електромашинобудування та електрообладнання. Міжвідомчий науково-технічний збірник. - 2008. - № 70. - Київ. : "Техніка". С. 28-32.
7. Гороховський О.І., Трояновська Т.І. Моделювання, створення та практика автоматизованих систем дистанційного навчання // Інформаційні технології та комп'ютерна інженерія. - 2007. - С. 235-239.

Гороховський Олександр Іванович, к. т. н., доцент кафедри обчислювальної техніки, тел.: (0432) 59-84-05, e-mail: goroh@vstu.vinnica.ua.

Трояновська Тетяна Іванівна, аспірант, e-mail: trtet@mail.ru.
Вінницький національний технічний університет.